**Meeting Notes 5/30/19**

-For the previous week’s assignment, Matt read tasks 1-6 and Paul read tasks 7-12. The short summary was also discussed that was put together with collaboration. A series of questions we had about Roofline were answered along the way.

-Roofline measures how far away you are from the achievable performance peak.

-Memory affinity is only on the chip cache. This was clarified when Paul asked if it applied to local memory accessed via bus or DMA. Also, all chips use cache.

-Location under the roof tells if you are limited by bandwidth or computational capacity.

-A memory bound kernel may stay memory bound after an optimization. This may be because the augmentation was not significant enough or misinterpreted.

-Order of steps of optimization is dependent on hardware implementation and other factors, not necessarily always in the same order.

-Array padding is done in C, not assembly. Higher level languages are desired as they are easier for the programmer and utilize the compiler. malloc is array padding. It is oversizing arrays with lines of length not a power of two to prevent repeat allocation. This leads to less chance of allocation to the same cache line. It also spreads memout out among available cache space. It can be thought of as trying to trick the allocation function to not repeat itself when using similar alternating allocation.

-2D array lengths should also not be powers of two to avoid overwriting.

-A Snoop filter supervises memory requests. It keep track of that frequency/periodicity to make prefetching recommendations.

-MVP (Matrix Vector Product) = O(5) = Constant run time. Not dependent on problem size.

-FFT (Fast Fourier Transform) is better than O(nlog(n)) but is problem size dependent. Increasing problem size may increase computational intensity.

-SSE is instructions to assembly in C. They explicitly tell the compiler to do something. As compilers improve, this concept will not be needed. It is currently a good step for the architect to go back to the basics for optimization.

**Discussion of the bmc article**:

-This project hopes to improve the performance of piRNA, BPMax and BPPair algorithms.

-We will focus on the implementation of the algorithms with roofline strategies. The algorithms will stay the same.

-RNA folding on itself 1x = Si,j (This is clarification for the assignment given for this week)

-An Eddy diagram is a graphical expression code. Dr. Rajopadhye demonstrated the correlation on the whiteboard. An email was also shared regarding the rules of the diagram lines and what the mean.

-In the questions portion of the meeting, it was clarified that we should begin understanding the algorithms and focusing on building our understanding of an implementation.

-At the end of the discussion we were asked the main points that we learned. Our answers included the wide variety of optimization strategies, the accuracy that one can model a computer and a kernel’s performance with, and the importance of baseline algorithms.

-An important point brought up by Dr. Rajopadhye is to research with efficiency as to not become overwhelmed with high level material.

**Assignment (Complete by the Night Before Next Meeting)**:

320 textbook dynamic programming chapter - Matrix Chain Product (MCP) - complete an Eddy diagram.

-It is noted Paul has not read this yet, but Matt has.

What does it mean to move up, right, left and down under the flat part and oblique part of the Roofline model?

What is the operational intensity of the MCP dynamic programming algorithm in the textbook (of the algorithm)?